REMARKS/ARGUMENTS

In view of the Amendments made to the claims, and in view of the following remarks, reconsideration of the application is respectfully requested.

1. Priority/Drawings

Applicants hereby note with appreciation that an acknowledgment for priority has been made in the last Office Action and that the formal drawings have been approved.

2. Prior Rejection under 35 U.S.C. §102(b)

Applicants hereby note with appreciation that the previous rejections under 35 U.S.C. § 102(b) have been withdrawn.

3. Claim Rejections under 35 U.S.C. §102

Claims 21, 23-26, 36, 38 and 40 have been rejected under 35 U.S.C. § 102(e) as being anticipated by Rosenthal (U.S. Patent No. 5,073,933). This rejection is respectfully traversed.

The present invention relates to a method for improving security on a computer system wherein a process identifier is used to indicate to a user when a trusted path has been established. Essentially, the process identifier is associated with each user during the login process. This process identifier feature of the invention is useful if untrusted code can temporarily disable or delay the effect of activation of a secure attention key. For example, sometimes untrusted code can send a reset command to some terminals. While these terminals are resetting, activation of the secure attention key has no effect. Such a technique may allow untrusted code (presumably maliciously written by someone attempting to obtain unauthorized

access to the system) to trick the user into thinking a trusted path has been established when, in fact, it has not. In accordance with at least claims 21-26, 30 and 35-40, the process identifier feature of the invention inhibits such trickery by displaying the process identifier to the user, allowing the user to distinguish between actual and emulated trusted paths.

As shown in Figure 6 of the subject application, when a user initially logs on to the computer system through the secure server (SSVR) 12, a randomly generated process identifier is assigned to the user. This activity is represented by step 310. In one embodiment, this process identifier comprises alphabetic characters. The process identifier is then stored in trusted memory by the SSVR 12 and displayed to the user by the SSVR 12 at step 320. If the trusted computing base (TCB) determines the trusted path has not been established, the process identifier is not displayed to the user. Each time a trusted path is established between the SSVR 12 and the user, the SSVR 12 displays the user process identifier at step 340. By observing the displayed process identifier, the user is assured that an actual trusted path has been established.

By contrast, U.S. Patent No. 5,073,933 is directed to an X Window Security System running on a server and a host computer terminal for allowing users to view only resources of the X Window server system which have been specifically authorized, thus rendering the system secure. In the past, in an X Window environment, users of various authorized host computers could use the server and all its resources. The patent attempts to change the security system from a host-based authentication system, wherein anyone on a particular host machine may access the system, to a user-based authentication system which allows access to be restricted to individuals rather than to computers. In order to implement this system, the patent proposes that a "NetName" be provided for each particular user and the NetName be the name by which a user is addressed within the X Window System. Since the NetName is proposed to be arranged as a new address family, it may be addressed like other address families called by the X Window System. Therefore, the "change host" command could be use to add and remove particular NetNames from the

authorized list and the "list host" command can be used to examine the names on the list. By providing NetName as a new address family, it is not necessary to add additional commands to the X Window System to access NetNames. When a user logs onto a host computer, the user password is used to decrypt to the user's secret key which is then stored on the host computer. To contact the server, a credential encrypted with the server's public key and containing the NetName of the client and a verifier including a time stamp is constructed. The verifier is encrypted using the client's secret key. When the server receives the credential, an authorization mechanism is employed to receive the credential and its secret key to decrypt the credential. The authorization mechanism then uses the client's public key to decrypt the verifier. Presuming the verifier can be decrypted by the client's public key, with the server only knowing that the user could have generated that credential. The authorization mechanism then checks the credential against a network-wide database to determine the authorization of that particular user. Essentially, the authorization mechanism is implemented in the server along side the host-based mechanism that takes the credential from the client, decrypts it to obtain the NetName of the user running the client and compares the NetName with the NetNames on the authorized list. If there is no comparison, the connection is refused. This same authorization mechanism also maintains the list of authorized NetNames. It adds and deletes entries when a "change host" request with the family name NetName is encountered. The authorization mechanism also returns entries from the list on receipt of a "list host" request.

With reference to claim 21, step (a) as previously presented refers to, upon login by a user, assigning a process identifier to the user in a trusted computing environment. The Examiner has argued on page 4 of the Office Action that this particular limitation is anticipated by Rosenthal, and he explains his position as follows: "(a) upon login by a user (see column 4, lines 17-19; when a user logs on; see column 3, lines 25-27; on a client on the local host (physically at the server) to change the access control list), assigning a process identifier to the user in the trusted competing environment (see column 4, lines 57-67; column 5, lines 1-3; adding an

entry of a new authorized NetName identifying a process forming a session between a user and a server)." Presumably, the Examiner is arguing that, if a user intended to log on to the X Window System and change the access control list, presumably a user would log on to a host with a password authorized by the authorization mechanism and then send the "change host" command to change the NetName. Thus, with the entry of a new authorized NetName, the NetName would be the process identifier identifying a process forming a session between the user and the server.

There are two problems with this general argument. First, the claim language reads "**upon** login by a user, automatically assigning a process identifier to the user in a trusted computing environment" (emphasis added). In other words, in this case, something occurs upon log on by a user. The prior art shows no such assigning of a process identifier upon log on. Rather, a user must log onto the X Window System and then, once the user has been authorized, a "change host" command is sent to then change the NetName. It should be noted that, to more explicitly claim the subject matter, claim 21 has been amended to indicate that the assigning of a process identifier occurs automatically upon login. Regardless, Applicant respectfully submits that the NetName disclosed in Rosenthal cannot reasonably be construed as a process identifier as recited in claim 21 and therefore, this claim should be allowed.

Second, the Examiner has not identified what would be considered the trusted computing environment. The trusted computing environment recited in claim 21 refers to a very specific type of software running within a computer itself. Applicants respectfully submit no such trusted computing environment as claimed is present in the applied prior art. Quite the contrary, according to the prior art, the X Window System was designed for use in a university environment in which the overhead in sophisticated research protection mechanisms was considered inappropriate. One problem with the X window system is that it is insecure at least in two respects, i.e., there is no control over who may access the resources of the system nor over what they may do with the resources once access has been gained. See column 1, lines 41-63. Even with the security modifications proposed by the patent, there is no trusted

computing base. Presumably the Examiner is reading all of the software running on the server as being the trusted computing environment. This is totally contrary to the claims of the invention wherein the trusted computing environment is considered a small, thus reliable and stable, computing space that has relatively few lines of code.

The Examiner has alleged that paragraph (e), which recites upon the user's subsequent entry into the computing environment, automatically displaying the process identifier to the user through the trusted path so that the user is assured that the trusted path has been established, is anticipated by a user who is logged on and is allowed to talk to the server which informs the host that the user has successfully logged on by displaying the NetName on the monitor to the user. Applicant respectfully submits that no such disclosure can be found in the prior art. Nowhere is it disclosed that the NetName is automatically displayed on the monitor to the user upon login. Quite the contrary, relevant language in column 1, lines 55-56, refers to a standard X Windows System that has not been modified by a security system. In that environment, the patent teaches that any intruder that has gained access to any of these computers, meaning computers in that network, then has unrestricted access to the server. For example, the patent specifically states the intruder may observe a user's keystrokes, button pushes, make mouse motions and monitor any output visible on the display. This portion of the patent does not indicate that the NetName is shown on the display at all, let alone upon login. Indeed, when a user logs onto a host, as described in column 4, lines 17+, the user password is used to decrypt a secret key and store it on the host computer. If a process client wishes to contact the server, it constructs a credential encrypted with the server's public key and containing the NetName of the client. When the server receives the credential, the server uses its secret key to decrypt the credential and then uses the client's public key to decrypt a verifier. The credential is checked against a network-wide database to determine the authorization of a particular user. Nowhere does the patent disclose automatically displaying the NetName on the display of the user. Even if a "list host" request is sent to the server and the NetName address is returned, this does not imply that such

NetName addresses are assigned upon login by the user and automatically displayed upon a subsequent login by the user.

As for the dependent claims 23-26, 36, 38 and 40, although they have further distinguishing features, they should at least be considered allowable by virtue of their dependency.

4. Rejection under 35 U.S.C. §103 of Claims 22, 35, 37 and 39

Claims 22, 35, 37 and 39 have been rejected under 35 U.S.C. § 103 as being unpatentable over Rosenthal (U.S. Patent No. 5,073,933) as applied to claim 21 in further view of Atalla (U.S. Patent No. 4,315,101). The Examiner has argued that Atalla teaches a random process identifier produced each time it is used. Thus, according to the Examiner, in both the methods of Rosenthal and Atalla, the process identifier is unique each time it is used. Therefore, the Examiner argues that it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the method of Rosenthal with the random process identifier of Atalla to prevent replay attacks. However, Applicants respectfully submit that, in the present invention, the process identifier is created once a user connects to a trusted computer base and that later on, when the user reconnects, that same, unchanged process identifier is automatically displayed to the user to indicate a trusted path has been formed. Note for example the "verifier" of Rosenthal is never displayed nor is the "user identifier code" of Atalla. In other words, even if Rosenthal and Atalla were hypothetically combined as set forth by the Examiner, the limitations of claims 22, 35, 37 and 39 would not be met and therefore, Applicants respectfully submit these claims should be allowed.

5. Rejections under 35 U.S.C. §103 of Claims 27 and 29-32

Claims 27 and 29-32 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Rosenthal (U.S. Patent No. 5,073,933) in view of Rivest et al. (U.S.

Patent No. 4,405,828). This rejection is respectfully traversed for at least the following reasons.

Initially, the Examiner is requested to examine these claims in accordance with 35 U.S.C. § 112, sixth paragraph, means plus function format. The claims include several such limitations. The untrusted parsing means for generating a trusted parse command recited in claim 27 essentially is a parser residing in the general untrusted operating system 20 which may parse a trusted command strain and convert it to a binary representation. Parsers typically check syntax and semantics and prompt for missing parameters. While parsing strategies were well-known in the art at the time of the invention, the use of an untrusted parser in trusted software systems was unknown. Due to the complex nature of parsing, large amounts of computer code are generally associated with this activity. Prior art trusted software code systems have included a parser for trusted commands and thus place the code within the trusted computing base. In these systems, every trusted command was parsed and executed exclusively by trusted code. The inclusion of the parsing code in the trusted computing base was required as necessary for proper system operation. See supporting disclosure, page 34, line 18-24, page 35, lines 1-5 and page 9, line 18 through page 10, line 5 of the Applicant's specification.

While there are many differences between the current claim 27 and the prior art, the most clear one perhaps is in paragraph (c) concerning a means for displaying a representation of the trusted and encrypted command to the user for verification. According to the Examiner, Rosenthal in column 5, lines 1-3 discloses that a user who is logged on is allowed to talk to the server which informs the host that the user has successfully logged on displaying the NetName on the monitor to the user. Applicant strongly traverses this position. Nowhere in column 1, lines 55-56 is the NetName disclosed to a user. Rather, the discussion in column 1, lines 55-56 refers to an X Window System unmodified by the patent's disclosed security system. The patent only discloses displaying the NetName to a user in response to a "list host" command, never for verification of a trusted encrypted command. Additionally, in reference to

paragraph (d) concerning a means for executing the verified trusted parse command, the Examiner refers again to column 5, lines 1-3 allowing the user who is logged on to talk to the server. However, one must remember that the verified trusted parsed command, according to the Examiner, is "NetName" as discussed above in regards to paragraph (c). Since NetName is actually not a command, it is unclear how the Examiner is reading the server as executing the same. For example, what exactly does the server do upon receiving the command "NetName"? Since NetName is not a command, the server cannot do anything and therefore, the limitation of a "means for executing the verified trusted parsed command" is not met by the prior art.

In regards to claims 31 and 32, these claims have been amended to more clearly set forth an untrusted parsing means for generating a parsed trusted command. As such, these claims specifically require an untrusted parsing means for generating a parsed trusted command and the means readable by the machine for causing the machine to receive the trusted parsed command from the untrusted parsing means. The Examiner has argued that Rosenthal discloses an apparatus for controlling the execution of a trusted command because the possibility of sending out a command "change host" which will add or remove particular NetNames from the authorized lists. However, the Examiner's attention is directed to column 4, lines 51-67 which disclose that an authorization mechanism is implemented in a server that takes a credential from a client, decrypts it to obtain a NetName of the user running the client, and compares the NetName with NetNames on the authorized list and, if there is no comparison, the connection is refused. In other words, an authorization mechanism is the mechanism which determines whether or not a user may log onto the server. The next paragraph indicates **that same authorization means** also maintains a list of authorized NetNames and adds and deletes entries when a "change host" request with the family name NetName is encountered. Applicants respectfully submit that the Examiner can either characterize the authorization mechanism as trusted or untrusted, but not both. Therefore, Applicants respectfully submit that sending a "change host" command, as taught by Rosenthal, even when modified by Rivest, does not anticipate the limitations of the claim.

In an apparent alternative argument, the Examiner appears to read the host as being an untrusted environment and the server as being a trusted environment. In effect, the Examiner is reading the log on command as being the trusted parsed command. However, according to Rosenthal, when a process or client wishes to contact the server, a credential is created, encrypted with the server's public key containing the NetName of the client and a verifier. Such a credential could not reasonably be considered parsed. In fact, quite the opposite, it is encrypted. It is the server which receives the credential that must use its secret key to decrypt the credential. See column 4, lines 17-32. This of course means the server, which in this case the Examiner is reading as a trusted means, must do the parsing. The claims however require that the **untrusted parsing means do the parsing** and therefore, they are not met by this prior art.

6. Allowable Subject Matter

Applicant hereby note with appreciation that the Examiner has indicated that the subject matter of claims 33 and 34 to be allowable. Please note that claim 41, which depends on claim 27 has been added and generally contains the disclosed limitations which the Examiner pointed out as allowable in claims 33 and 34. Therefore, presumably this claim should also be considered allowable.

In view of the amendments made to the claims, the discussions and agreements made with the Examiner, and the above remarks, allowance of this application is respectfully requested. If the Examiner should have any concerns regarding this Amendment/Response, he is cordially invited to contact the undersigned at the number provided below.

Respectfully submitted,

Nicholas S. Whitelaw
Attorney for Applicants
Registration No. 36,418

Date: February 11, 2005
DIEDERIKS & WHITELAW, PLC
12471 Dillingham Square, #301
Woodbridge, VA 22192
Tel: (703) 583-8300
Fax: (703) 583-8301